VAE-SMOTE Augmented Diffusion for Anomaly Detection

Lei Feng
School of Software
Xinjiang University
Urumqi, Xinjiang, China
107552204750@stu.xju.edu.cn

Jiong Yu*
School of Information Science and Engineering
Xinjiang University
Urumqi, Xinjiang, China
yujiong@xju.edu.cn

Shu Li
School of Information Science and Engineering
Xinjiang University
Urumqi, Xinjiang, China
lsmiao@stu.xju.edu.cn

Shicheng Jiu
School of Software
Xinjiang University
Urumqi, Xinjiang, China
jiusc@stu.xju.edu.cn

Abstract

Most current generative models for anomaly detection, which rely on reconstruction methods, primarily use either Variational Autoencoders or Generative Adversarial Networks. However, VAEs tend to suffer from poor reconstruction quality, while GANs face challenges related to training stability and limited pattern coverage. Additionally, due to privacy and security concerns, the available training data for anomaly detection is often limited, whereas generative models typically require large datasets to avoid the risk of overfitting. To overcome these challenges, we introduce an unsupervised anomaly detection model utilizing diffusion techniques, designed specifically for tabular data, using a reconstruction-based approach. We adopt the diffusion model as the backbone for anomaly detection, aiming to enhance reconstruction quality and ensure training stability. To further improve reconstruction quality and accelerate inference, we employ a truncated diffusion strategy during inference. What's more, to overcome the challenge of insufficient training data, we introduce a SMOTE-based data augmentation method optimized for feature space in tabular data, incorporating a VAE to address the limitations of the original SMOTE algorithm in capturing complex data structures and handling nonlinear relationships. Experimental results on real-world datasets validate that our method achieves higher detection accuracy compared to conventional methods, demonstrating its effectiveness.

CCS Concepts

• **Computing methodologies** → Machine learning; Learning paradigms; Unsupervised learning; Anomaly detection.

Keywords

Anomaly Detection, Data Augmentation, Diffusion Model

*Corresponding author.

GAIIS 2025, Hangzhou, China

ACM ISBN 979-8-4007-1345-3/2025/02
https://doi.org/10.1145/3728725.3728751

ACM Reference Format:
Lei Feng, Jiong Yu, Shu Li, and Shicheng Jiu. 2025. VAE-SMOTE Augmented Diffusion for Anomaly Detection. In *2025 2nd International Conference on Generative Artificial Intelligence and Information Security (GAIIS 2025), February 21–23, 2025, Hangzhou, China.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3728725.3728751

1 Introduction

Anomaly detection identifies outliers deviating from normal patterns[3], with applications in healthcare, finance, and remote sensing. Supervised methods rely on labeled data, but obtaining anomaly labels is challenging. Unsupervised methods, which model normal data distributions to flag deviations, are more practical. Traditional methods like KNN and LOF struggle with high-dimensional and non-linear data, while model-based methods such as Isolation Forest are sensitive to noise.

Deep learning excels in feature extraction from complex data, with methods like contrastive learning and graph neural networks (GNNs) enhancing performance but often requiring extensive training or high-quality graphs. Reconstruction-based methods offer an alternative, requiring less training time and assuming no fixed data distribution. In this field, Variational Autoencoders (VAEs)[10] and Generative Adversarial Networks (GANs)[5] are representative generative models, which detect anomalies by reconstructing normal data and analyzing reconstruction errors. Normal samples produce low reconstruction errors, while anomalies deviate significantly.

However, VAEs may yield low-quality reconstructions [4], and GANs encounter problems such as mode collapse. Denoising Diffusion Probabilistic Models (DDPMs) [9] address these challenges with stable training, high pattern coverage, and resistance to mode collapse. While diffusion models excel in generating high-quality image samples, their application to tabular anomaly detection is underexplored.

Generative models require extensive data but often face overfitting with limited datasets common in anomaly detection. Data augmentation techniques, including VAE-based (TVAE) and GAN-based (CTGAN) methods, address this. Traditional SMOTE [2] shows potential but struggles with complex, high-dimensional data due to its simplistic linear interpolation.

This paper proposes a diffusion model combined with SMOTE-enhanced data augmentation for tabular anomaly detection. We

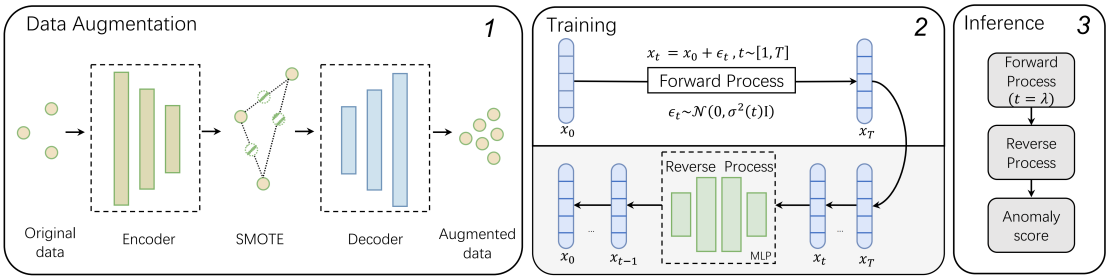

Figure 1: The proposed VADAD model. VADAD consists of three phases: (1) Data augmentation on the training set, (2) Training the diffusion model, and (3) Using the trained diffusion model for anomaly detection.

apply the SMOTE algorithm in the latent space of Variational Autoencoders (VAEs) to better capture nonlinear features. After generating new samples, the decoder is used to map these samples back to the original data space. The diffusion model, trained on the enhanced dataset, increases normal sample diversity and improves reconstruction quality, amplifying the reconstruction error gap between normal and anomalous data.

Overall, our paper makes the following key contributions:

- We introduce a new approach for anomaly detection in tabular data, VADAD, which combines data augmentation and diffusion model to significantly enhance detection effectiveness.
- We propose a feature-space optimization-based method called VSA for tabular data augmentation, which integrates SMOTE and VAE. By enhancing the scale and diversity of original training data, the diffusion model can more effectively learn the distribution characteristics of normal samples.
- We perform extensive experiments on 18 real-world tabular datasets spanning multiple domains. Our method exhibits superior detection accuracy over seven advanced anomaly detection algorithms, as evidenced by the results.

2 RELATED WORK

Traditional methods primarily include distance-based, model-based, clustering-based, and statistical techniques. However, these approaches often struggle with complex data, paving the way for the adoption of deep learning solutions.

Deep learning offers superior performance in anomaly detection. Methods such as LUNAR [6], ICL [13], and DIF [14] significantly improve the performance of anomaly detection compared to traditional methods. Reconstruction-based methods identify anomalies using reconstruction similarity metrics $S(x, x^{'})$, where $x^{'} = \mathcal{R}(x)$ is reconstructed from x. Models trained on normal data reconstruct normal samples well, while anomalies yield high errors. GANomaly [1] employs an encoder-decoder-encoder generator and adversarial training but faces unstable training and limited pattern coverage. MCM [17] uses multiple masks in autoencoders for reconstruction error detection, though autoencoder limitations reduce accuracy. DDPM [9] and DTPM [12] leverages diffusion models for high-quality reconstructions, enabling effective anomaly detection. Similarly, our approach uses diffusion to enhance anomaly detection.

Synthesizing high-quality tabular data has gained attention with methods like TVAE, CTGAN [16]. SMOTE, a traditional approach, generates new samples by interpolating between a sample and its neighbors, effectively balancing datasets. However, it struggles with non-linear structures. Our method, VADAD, enhances SMOTE by incorporating VAE to better capture complex relationships, improving the quality and diversity of generated data.

3 MeTHOD

3.1 Overview

Figure 1 illustrates our proposed VADAD method. First, a subset of normal samples is extracted from the original dataset to form the training set $\mathcal{D}$, which is used to train a VAE. After mapping the training set to the latent space, the SMOTE algorithm generates new samples with similar characteristics, thereby enhancing both the scale and diversity of the data. These samples are decoded back into the original space, preserving essential features and distributions. The expanded training set is then used to train a diffusion model, improving its ability to learn normal data patterns. Finally, during anomaly detection, reconstruction errors between test samples and their reconstructed counterparts are used to identify anomalies.

To improve reconstruction quality and accelerate inference during the testing phase, we employ a truncated diffusion approach, where the forward diffusion process is stopped early. The truncation point is selected at λ (where $0 < \lambda < T$)

3.2 Data Augmentation

Data augmentation consists of VAE and SMOTE methods. The encoder and decoder are trained using a dataset $\mathcal{D}_{OK} \in \mathbb{R}^{N \times p}$, which consists of normal samples extracted from the original dataset. The loss function comprises two components, with the first being the reconstruction error of the samples:

$$\mathcal{L}_{recon} = \frac{1}{N} \sum_{i=1}^{N} ||x_i - \hat{x}_i||^2 \tag{1}$$

Here, the number of samples is denoted by N, the i-th input sample is represented by x_i, and its corresponding reconstructed version is denoted by $\widehat{x_i}$.

In the second loss term, the difference is measured using the Kullback-Leibler divergence. The KL divergence loss, denoted as $\mathcal{L}_{kl}$, quantifies the difference between the latent space distribution $q(z|x)$ generated by the encoder and the distribution p(z), where $p(z) \sim \mathcal{N}(0,1)$. $\mathcal{L}_{kl}$ is formulated as follows:

$$\mathcal{L}_{kl} = -\frac{1}{2} \cdot \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{p} \left(1 + \log\left(\sigma_{ij}^2\right) - \mu_{ij}^2 - \sigma_{ij}^2\right) \tag{2}$$

where p is the latent space dimension, μ_{ij} is the mean of the j-th latent variable for the i-th sample; $\log(\sigma_{ij}^2)$ is the logarithmic variance; and σ_{ij}^2 is the variance, computed as the exponent of the logarithmic variance; The negative sign is introduced to convert the KL divergence term into a loss term, which can then be minimized.

The KL divergence term ensures that the latent space distribution of the VAE approximates a standard normal distribution, facilitating feature space optimization and resulting in a more organized latent structure. This regularization ensures that the distribution of sample points is smoother, with similar input samples having closer representations[18]. Consequently, interpolating between or generating new samples in this optimized latent space captures the data's feature distribution more accurately, resulting in higher-quality sample generation and enhanced data augmentation. The joint loss is defined as a weighted sum of the two loss terms from Equations 1) and (2):

$$\mathcal{L} = w_1 \mathcal{L}_{recon} + w_2 \mathcal{L}_{kl} \tag{3}$$

Since the SMOTE approach is a non-parametric method, only the VAE needs to be trained. During the training process, $Encoder(\cdot)$ and $Decoder(\cdot)$ are co-optimized as a whole. The data augmentation process starts after the VAE training is completed. Algorithm 1 shows the complete data augmentation process.

Algorithm 1 Data Augmentation

Input: Original normal dataset $\mathcal{D}_{OK} \in \mathbb{R}^{N \times p}$ (extracted from the original dataset), Well-trained $Encoder(\cdot)$ and $Decoder(\cdot)$
Output: Augmented normal dataset $\mathcal{D}_{OK}^*$
1. $M = 1.5 \times N$
2. $Z_{OK} = Encoder(\mathcal{D}_{OK})$, $where\ Z_{OK} \in \mathbb{R}^{N \times d}$
3. $X_{random} \sim \mathcal{N}(0,1)^{M \times d}$, $y_{NG} = [1, 1, \ldots, 1] \in \mathbb{R}^M$
4. $Z_{NG} = \{(x_i, y_i) | x_i \in X_{random}, y_i \in y_{NG}\}$
5. $Z = Z_{NG} \cup Z_{OK}$
6. $Z_{gen} = SMOTE(Z)$
7. $\mathcal{D}_{aug} = Decoder(Z_{gen})$
8. $\mathcal{D}_{OK}^* = \mathcal{D}_{aug}(y = 0)$, $\mathcal{D}_{OK}^* \in \mathbb{R}^{M \times p}$
9. **return** $\mathcal{D}_{OK}^*$

3.3 Diffusion Modeling

Diffusion model consist of two primary stages: the forward diffusion stage and the reverse reconstruction stage.

3.3.1 Forward Process. The step-by-step addition of noise ϵ to the forward process produces a series of potential noise variables $x_1, \ldots, x_T$ sharing the dimensionality of the input data $x \sim q(x)$. These T steps can be represented by equations 4):

$$q\left(x_t | x_{t-1}\right) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t} x_{t-1}, \beta_t \mathrm{I}\right) \tag{4}$$

Noise is applied step by step. Specifically, the variance schedule $\beta_t \in (0,1)$ determines the amount of noise. This schedule can be configured in various ways; in our approach, the schedule follows a linear progression, starting at $\beta_1 = 10^{-4}$ and ending at $\beta_T = 0.02$.

3.3.2 Reverse Process. The reverse process is a sampling procedure implemented through a network model parameterized by θ, which we denote as $\mu_\theta(\cdot)$. The sampling from $x_T \sim \mathcal{N}(0, \mathrm{I})$ can be expressed as:

$$p_\theta\left(x_{t-1} \mid x_t\right) = \mathcal{N}\left(x_{t-1} | \mu_\theta\left(x_t\ , t\right), \tilde{\beta}_t \mathrm{I}\right) \tag{5}$$

Here $t = 1, \ldots, T$ and $\widetilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t$. The architecture of μ_θ is typically implemented using either a U-Net or ResNet structure. However, in our method, we employ an MLP structure, as it is more suitable for handling tabular data. The MLP structured as follows:

$$MLP\left(x\right) = Linear(ReLU\left(...ReLU\left(Linear\left(x\right)\right)\right) \tag{6}$$

We make $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=0}^{T} \alpha_i$, equation 4) can be described as:

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t} \epsilon_t\ , \quad \epsilon_t \sim \mathcal{N}\left(0\ , \mathrm{I}\right) \tag{7}$$

We consider the representation of $\mu_\theta(x_t, t)$ using the following equation to achieve better training results:

$$\mathcal{L}_{MSE} = \mathbb{E}_{t \sim [1-T], x_0 \sim q(x_0), \epsilon \sim N(0,\mathrm{I})}\left[\epsilon - \epsilon_\theta(x_t, t)^2\right] \tag{8}$$

3.4 Inference

By progressively adding noise to the original data x_0, a series x_t is generated with decreasing similarity to x_0. Graham *et al*[7] found that up to a certain time step t' denoised data retains high similarity with x_0, but beyond t', the denoised result may diverge significantly, potentially generating unrelated samples. While useful for enhancing diversity in image generation, anomaly detection requires reconstructed samples to closely resemble the original, making it essential to retain sufficient information from x_0 to avoid large reconstruction errors.

In the anomaly detection stage, we adopt a truncated diffusion method, allowing the model to halt the forward diffusion process partway through. We choose to truncate the process at λ (where $0 < \lambda < T$).

4 Experiment

4.1 Experimental Settings

4.1.1 Dataset. We selected 18 tabular datasets commonly used in anomaly detection, spanning various application domains, as shown in Table 1. These datasets were sourced from ADBench[8].

4.1.2 Evaluation Metrics. Normal and abnormal data were randomly split 80/20; 80% of the normal data was for training, while the rest, along with 20% of the abnormal data, was used for testing. Data was shuffled to avoid order bias[11]. We used two complementary metrics—Area Under the ROC Curve (AUC-ROC) and Area Under the Precision-Recall Curve (AUC-PR)—to evaluate detection accuracy. The ROC curve depicts the connection between the

Table 1: Details of datasets used. The "Feat." indicates the features, while the "Anom." column lists the anomaly percentages.

Dataset	Samples (Feat.)	Anom. (%)	Dataset	Samples (Feat.)	Anom. (%)
Hepatitis	80 (19)	16.25	Fault	1,941 (27)	34.67
Wine	129 (13)	7.75	SpamBase	4,207 (57)	39.91
WBC	223 (9)	4.48	M.gamma	5,000 (10)	53.89
Vertebral	240 (6)	12.5	ALOI	5,000 (27)	2.88
Ionosphere	351 (32)	35.9	Backdoor	5,000 (196)	2.48
WDBC	367 (30)	2.72	Fraud	5,000 (29)	0.12
Breastw	683 (9)	34.99	Census	5,000 (500)	6.13
Optdigits	5,216 (64)	2.88	Satellite	6,435 (36)	31.64
Pendigits	6,870 (16)	2.27	Mnist	7,603 (100)	9.21

True Positive Rate (TPR) and the False Positive Rate (FPR). If the AUC-ROC score is higher, it implies that the model achieves superior performance in differentiating normal and anomalous samples. Meanwhile, the PR curve illustrates the relationship between Precision and Recall. Higher AUC-PR values, particularly in cases where anomaly samples are sparse, indicate that the model is more effective in performing anomaly detection.

4.1.3 Comparison Algorithms. We selected seven prominent baseline methods in the field of tabular anomaly detection for comparative analysis: LUNAR [6], ICL [13], DIF [14], GANomaly [1], DDPM[9], DTPM [12], and MCM [17]. The implementations of these methods were sourced from the PyOD [19], DeepOD [15] libraries or their officially released code. Additionally, GANomaly, DDPM, DTPM, and MCM share the same reconstruction-based approach to anomaly detection as our method. All algorithms were evaluated based on the same dataset partitioning and preprocessing conditions, with parameter settings consistent with their respective original papers.

4.1.4 Parameter Setting. During data augmentation, the encoder and decoder are symmetric three-layer MLPs with ReLU activation, using hidden dimensions of 256-128-64, and the embedding layer dimensionality is set to 16. The joint loss combines reconstruction error and KL divergence with weights of 1 and 0.2. A total of 5000 epochs were used for training, with a batch size set to 512 using Adam (lr: 2e-4, weight decay: 1e-3). SMOTE used k=7. During diffusion modeling, The noise prediction network is an MLP with the architecture d-128-256-256-128-d, where d is the input dimensionality. Diffusion used T=1000 and $\lambda = 200$, trained for 5000 epochs with batch size matching the training set size, and Adam optimizer (lr: 5e-3).

4.2 Results and Analysis

4.2.1 Experimental Results. This section assesses the performance of eight algorithms across 18 datasets. The outcomes of anomaly detection are summarized in Tables 2, with AUC-ROC and AUC-PR serving as the evaluation criteria. The top-performing results are bolded, with the second-best underlined. The results were conducted on a single RTX 4090 GPU.

4.2.2 Statistical Tests and Average Rankings. This section outlines the use of the Friedman test to statistically validate the performance differences between VADAD and the baseline methods across multiple datasets. Since relying on a single metric or dataset may not provide a comprehensive view, the Friedman test was applied as a nonparametric method to compare the average rankings of these methods. Table 3 summarizes the experimental results.

4.2.3 Visualization of Reconstruction Effects. To further analyze the performance of our method, we applied KDE (Kernel Density Estimation) plots to visualize and analyze the reconstruction effect. As shown in Figure 2, the reconstructed normal data closely aligns with its original distribution, whereas the reconstructed abnormal data shows significant deviation from its original distribution.

4.3 Ablation Experiments

In this section, we conduct ablation experiments to assess the contributions of various components in our approach. The following tasks were defined for comparison:

- Task A: Applies the base diffusion model exclusively for anomaly detection, with no data augmentation.
- Task B: Adopts data augmentation on the training set using the standard SMOTE method, followed by diffusion model training for anomaly detection.
- Task C: Enhances the training data using the VAE-SMOTE approach (referred to as VSA), then trains the diffusion model for anomaly detection.

We calculate the average AUC-ROC and AUC-PR values across three tasks on 18 datasets, and the results are presented in Table 4. As shown, data augmentation effectively boosts the model's performance in detecting anomalies. Furthermore, our proposed VSA method outperforms the original SMOTE approach in improving the model's detection accuracy.

4.4 Different Data Augmentation Strategies

Table 5 presents the average performance of various data augmentation strategies, evaluated using AUC-ROC and AUC-PR metrics. The outcomes demonstrate that our proposed VSA data augmentation strategy outperforms the widely-used VAE-based (TVAE) and GAN-based (CTGAN) methods.

4.5 Hyperparameter Analysis

We analyzed four hyperparameters and found that the results for both metrics showed similar trends across different experimental

Table 2: Performance metrics (AUC-ROC and AUC-PR) for eight algorithms across 18 datasets.

Dataset	GANomaly	LUNAR	ICL	DIF	DDPM	DTPM	MCM	Ours
ALOI	(0.5406, 0.0358)	(0.6645, 0.0674)	(0.5576, 0.0439)	(0.5667, 0.0366)	(0.5758, 0.0365)	(0.5560, 0.0361)	(0.5624, 0.0371)	(0.6892, 0.0899)
Fault	(0.5615, 0.4028)	(0.7998, 0.6837)	(0.7092, 0.5106)	(0.6536, 0.5021)	(0.6381, 0.4784)	(0.6993, 0.5529)	(0.664, 0.4550)	(0.8002, 0.6741)
Fraud	(0.9990, 0.5000)	(0.9990, 0.5000)	(0.9970, 0.2500)	(0.9940, 0.1429)	(1.0000, 1.0000)	(0.9970, 0.2500)	(0.9950, 0.1667)	(1.0000, 1.0000)
Hepatitis	(0.0769, 0.1099)	(0.7692, 0.3333)	(0.8462, 0.6667)	(0.6923, 0.2667)	(0.3462, 0.1458)	(0.6538, 0.2500)	(0.7692, 0.3250)	(1.0000, 1.0000)
Iono-sphere	(0.8604, 0.7956)	(0.9751, 0.9566)	(0.9769, 0.9581)	(0.8409, 0.8300)	(0.8649, 0.8218)	(0.9004, 0.8932)	(0.9298, 0.8980)	(0.9716, 0.9600)
M.gamma	(0.5878, 0.4194)	(0.8560, 0.8102)	(0.7931, 0.7542)	(0.7362, 0.6703)	(0.7229, 0.6517)	(0.8580, 0.8066)	(0.7916, 0.7348)	(0.8178, 0.7695)
Mnist	(0.7625, 0.3169)	(0.9399, 0.6796)	(0.8990, 0.5395)	(0.8512, 0.4044)	(0.8362, 0.3938)	(0.9273, 0.5639)	(0.9098, 0.5463)	(0.9321, 0.6289)
Optdigits	(0.6226, 0.0365)	(0.9980, 0.9252)	(0.9474, 0.2452)	(0.5591, 0.0304)	(0.6492, 0.0439)	(0.8220, 0.0776)	(0.9333, 0.1831)	(0.9816, 0.5501)
Pendigits	(0.8564, 0.0763)	(0.9994, 0.9677)	(0.9534, 0.4272)	(0.9721, 0.5102)	(0.8487, 0.1170)	(0.9750, 0.3132)	(0.9902, 0.6098)	(0.9949, 0.7305)
Backdoor	(0.9454, 0.7037)	(0.9374, 0.8084)	(0.9258, 0.7707)	(0.8909, 0.5235)	(0.9066, 0.6575)	(0.9047, 0.6131)	(0.9326, 0.7503)	(0.9784, 0.8638)
Satellite	(0.8471, 0.7951)	(0.8816, 0.8429)	(0.8581, 0.8349)	(0.8077, 0.7663)	(0.7924, 0.7602)	(0.8100, 0.7997)	(0.8141, 0.7461)	(0.8564, 0.8139)
SpamBase	(0.8030, 0.6997)	(0.8285, 0.7616)	(0.8543, 0.8047)	(0.5628, 0.4226)	(0.6405, 0.5031)	(0.8430, 0.7346)	(0.6617, 0.5413)	(0.8549, 0.8231)
Vertebral	(0.5397, 0.1545)	(0.7619, 0.3686)	(0.3651, 0.2563)	(0.5198, 0.1786)	(0.5714, 0.3737)	(0.6230, 0.3266)	(0.5635, 0.1616)	(0.9603, 0.8409)
Breastw	(0.9956, 0.9925)	(0.9940, 0.9880)	(0.9662, 0.8519)	(0.9923, 0.9840)	(0.9802, 0.9511)	(0.9874, 0.9657)	(0.9765, 0.9336)	(0.9952, 0.9911)
WBC	(0.9762, 0.5833)	(1.0000, 1.0000)	(0.7857, 0.1838)	(1.0000, 1.0000)	(0.9881, 0.8333)	(0.9524, 0.4167)	(1.0000, 1.0000)	(1.0000, 1.0000)
WDBC	(0.7887, 0.2823)	(0.9930, 0.8333)	(0.8169, 0.1005)	(0.9930, 0.8333)	(0.7746, 0.0839)	(0.9789, 0.5000)	(0.9648, 0.4167)	(1.0000, 1.0000)
Wine	(0.9783, 0.8333)	(1.0000, 1.0000)	(0.9130, 0.4500)	(0.9565, 0.7500)	(0.9565, 0.5833)	(0.9348, 0.7000)	(1.0000, 1.0000)	(1.0000, 1.0000)
Census	(0.7108, 0.1106)	(0.6576, 0.0862)	(0.6394, 0.0971)	(0.6820, 0.0925)	(0.6211, 0.0888)	(0.7161, 0.1062)	(0.6590, 0.0995)	(0.7431, 0.1498)
Average	(0.7474, 0.4360)	(0.8919, 0.7007)	(0.8225, 0.4859)	(0.7928, 0.4969)	(0.7619, 0.4735)	(0.8411, 0.4948)	(0.8399, 0.5336)	(0.9209, 0.7714)

Table 3: Friedman test results are presented. Superior performance is associated with a lower average rank, with the top results shown in bold.

	p-value (<0.05)	statistic	Algorithm	AvgRank	Algorithm	AvgRank
AUC-ROC	2.2174e-09	54.1293	GANomaly	5.6667	LUNAR	2.2778
			DTPM	4.6667	ICL	4.9444
			DDPM	6.0556	MCM	4.3889
			DIF	5.6667	Ours	1.5556
AUC-PR	6.9222e-10	56.6794	GANomaly	6.0000	LUNAR	2.2222
			DTPM	4.7778	ICL	4.5000
			DDPM	6.0556	MCM	4.7222
			DIF	5.5000	Ours	1.5000

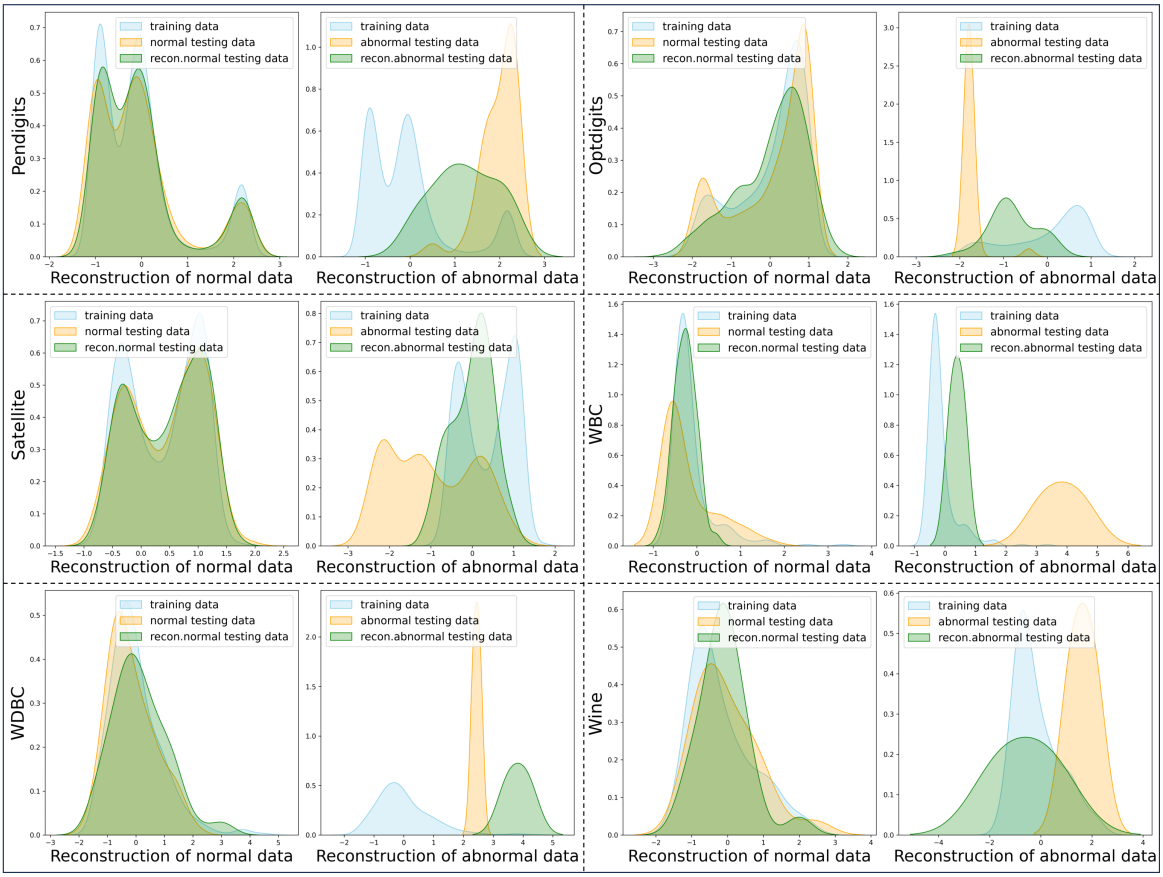

Figure 2: shows experimental results on six datasets, with two subfigures per dataset: the left for normal test data and the right for abnormal test data. Each subfigure compares three distributions: training data (blue), original test data (orange), and reconstructed data (green).

Table 4: Ablation study of VADAD

Task	VAE	SMOTE	diffusion	mAUC-ROC	mAUC-PR
A	-	-	√	0.9012	0.7372
B	-	√	√	0.9070	0.7460
C	√	√	√	0.9209	0.7714

Table 5: Mean AUC-ROC and AUC-PR outcomes for various data augmentation strategies.

Metrics	TVAE	CTGAN	SMOTE	VSA(ours)
mAUC-ROC	0.9015	0.8914	0.9070	0.9209
mAUC-PR	0.7142	0.7086	0.7460	0.7714

settings. Therefore, we present only the AUC-ROC results. As shown in Figure 3, subfigures (a), (b), and (c) indicate that most datasets are not significantly affected by changes in these three hyperparameters. For subfigure (d), the experiments were conducted under the condition of a fixed T. The results demonstrate that for most datasets, performance improves as λ increases up to 200 but declines thereafter. This observation validates the effectiveness of our truncated diffusion strategy.

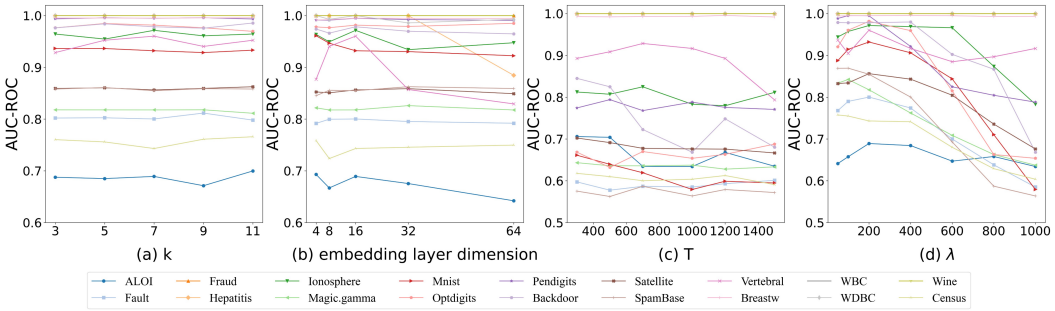

Figure 3: AUC-ROC of VADAD across 18 datasets with varying hyperparameter settings. (a) the number of nearest neighbors for sample interpolation in the data augmentation module, denoted as k, which comes from the SMOTE method; (b) the dimensionality of the embedding layer in the VAE, *emb*; (c) the full diffusion steps, T, used during training; and (d) the parameter λ for truncated diffusion during inference was experimented with under the condition that the T was fixed at 1000.

Table 6: Average runtime of eight methods on low- and high-dimensional datasets, along with their runtime gaps.

Dataset	GANomaly(s)	LUNAR(s)	ICL(s)	DIF(s)	DDPM(s)	DTPM(s)	MCM(s)	Ours(s)
Low_dim	0.0022	0.1511	0.0890	0.0029	0.0212	0.3214	0.0082	0.0279
High_dim	0.0032	0.3900	0.1154	0.0043	0.0201	0.5454	0.0115	0.0286
Gap	0.0010	0.2389	0.0264	0.0014	0.0011	0.2240	0.0033	0.0007

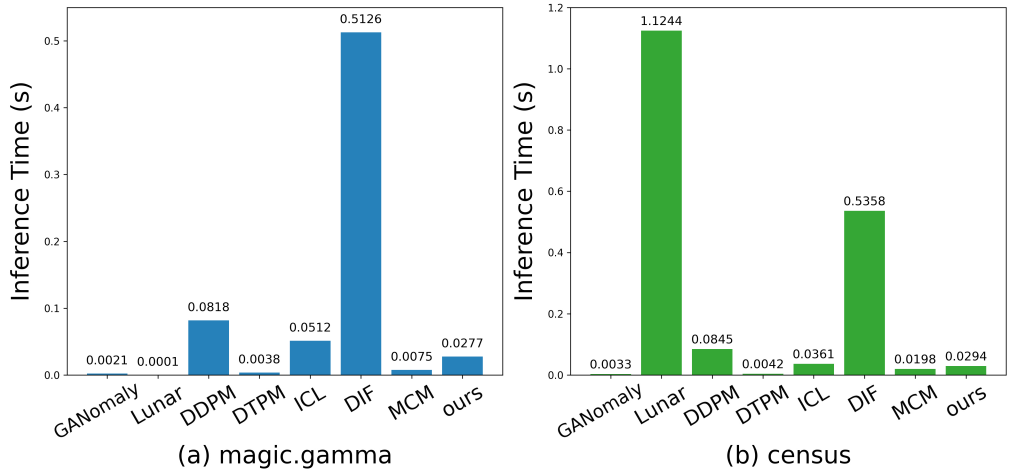

Figure 4: Inference times for eight models on a low-dimensional dataset (magic.gamma with 10 dimensions) and a high-dimensional dataset (census with 500 dimensions).The two datasets contain comparable amounts of data.

5 Computational Analysis

We define datasets with fewer than 50 dimensions as low-dimensional and those with more than 50 dimensions as high-dimensional. Table 6 presents the average runtime of different methods on both low- and high-dimensional datasets, along with the runtime gaps between them. The final row of the table demonstrates that our proposed method achieves the smallest runtime gap between the two types of datasets, showcasing its robustness. Notably, our method significantly improves anomaly detection performance. At the same time, it does not lead to a corresponding increase in inference time, demonstrating the efficiency of our approach.

We visualize the results on two representative datasets, as shown in Figure 4, our proposed method maintains a stable inference time across both high- and low-dimensional datasets.

6 Conclusions and Future work

In this paper, we adopt the diffusion model to conduct anomaly detection in tabular data using an unsupervised approach. To improve the diffusion model's capability in capturing the distribution of normal samples, we introduce a innovative data augmentation method, VSA, which integrates SMOTE and VAE. We evaluated VADAD on 18 benchmark datasets, and the results indicate that VADAD surpasses multiple state-of-the-art anomaly detection algorithms, achieving superior detection accuracy.

Future efforts will be directed toward further developing the methods introduced in this paper to accommodate different types of datasets, including time series and graph data. These data types differ from tabular data due to their inherent structural and feature-based complexities, which will require the development of new strategies to handle these challenges.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant Nos.62262064)

References

[1] Akcay, S., Atapour-Abarghouei, A., Breckon, T.P.: Ganomaly: Semi-supervised anomaly detection via adversarial training. In: Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14. pp. 622–637. Springer (2019).

[2] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research 16, 321– 357 (2002).

[3] Di Mattia, F., Galeone, P., De Simoni, M., Ghelfi, E.: A survey on gans for anomaly detection. arXiv preprint arXiv:1906.11632 (2019).

[4] Dosovitskiy, A., Brox, T.: Generating images with perceptual similarity metrics based on deep networks. Advances in neural information processing systems 29 (2016).

[5] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems 27 (2014).

[6] Goodge, A., Hooi, B., Ng, S.K., Ng, W.S.: Lunar: Unifying local outlier detection methods via graph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 6737–6745 (2022).

[7] Graham, M.S., Pinaya, W.H., Tudosiu, P.D., Nachev, P., Ourselin, S., Cardoso, J.: Denoising diffusion models for out-of-distribution detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2948–2957 (2023).

[8] Han, S., Hu, X., Huang, H., Jiang, M., Zhao, Y.: Adbench: Anomaly detection benchmark. Advances in Neural Information Processing Systems 35, 32142–32159 (2022).

[9] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems 33, 6840–6851 (2020).

[10] Kingma, D.P.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013).

[11] Li, S., Yu, J., Lu, Y., Yang, G., Du, X., Liu, S.: Self-supervised enhanced denoising diffusion for anomaly detection. Information Sciences 669, 120612 (2024).

[12] Livernoche, V., Jain, V., Hezaveh, Y., Ravanbakhsh, S.: On diffusion modeling for anomaly detection. arXiv preprint arXiv:2305.18593 (2023).

[13] Shenkar T, Wolf L. Anomaly detection for tabular data with internal contrastive learning[C]//International conference on learning representations. 2022.

[14] Xu, H., Pang, G., Wang, Y., Wang, Y.: Deep isolation forest for anomaly detection. IEEE Transactions on Knowledge and Data Engineering 35(12), 12591–12604 (2023).

[15] Xu, H., Wang, Y., Jian, S., Liao, Q., Wang, Y., Pang, G.: Calibrated one-class classification for unsupervised time series anomaly detection. IEEE Transactions on Knowledge and Data Engineering (2024).

[16] Xu, L., Skoularidou, M., Cuesta-Infante, A., Veeramachaneni, K.: Modeling tabular data using conditional gan. Advances in neural information processing systems 32 (2019).

[17] Yin, J., Qiao, Y., Zhou, Z., Wang, X., Yang, J.: Mcm: Masked cell modeling for anomaly detection in tabular data. In: The Twelfth International Conference on Learning Representations (2024).

[18] Zhang, H., Zhang, J., Srinivasan, B., Shen, Z., Qin, X., Faloutsos, C., Rangwala, H., Karypis, G.: Mixed-type tabular data synthesis with score-based diffusion in latent space. arXiv preprint arXiv:2310.09656 (2023).

[19] Zhao, Y., Nasrullah, Z., Li, Z.: Pyod: A python toolbox for scalable outlier detection. Journal of machine learning research 20(96), 1–7 (2019).